Fermi National Accelerator Laboratory

FERMILAB-Conf-87/76
2380.000

The ACP Branch Bus and Real-Time Applications of the ACP Multiprocessor System*

R. Hance, H. Areti, R. Atac, J. Biel, A. Cook, M. Fischler,
I. Gaines, D. Husby, T. Nash, and T. Zmuda

Advanced Computer Program
Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, IL 60510

May 8, 1987

*Invited talk at the Fifth IEEE Conference on Real-Time Computer Applications in Nuclear, Particle, and Plasma Physics, San Francisco, CA, May 12-14, 1987.

THE ACP BRANCH BUS AND REAL TIME APPLICATIONS OF THE ACP MULTIPROCESSOR SYSTEM

R. Hance, H. Areti, R. Atac, J. Biel, A. Cook, M. Fischler, I. Gaines, D. Husby, T. Nash, and T. Zmuda

Advanced Computer Program
Fermi National Accelerator Laboratory
Batavia, Il. 60510

Abstract

The ACP Branchbus, a high speed differential bus for data movement in multiprocessing and data acquisition environments, is described. This bus was designed as the central bus in the ACP multiprocessing system. In its full implementation with 16 branches and a bus switch, it will handle data rates of 160 MByte/sec and allow reliable data transmission over inter rack distances. We also summarize applications of the ACP system in experimental data acquisition, triggering and monitoring, with special attention paid to FASTBUS environments.

Introduction

The Advanced Computer Program (ACP) at Fermilab has designed a multiprocessor system for off line reconstruction of data from high energy physics experiments and high level on line triggers.[1] This system makes FORTRAN processing power available at a present cost of roughly $2000 per VAX 11/780 equivalent. The system is based on single board computers incorporating popular 32 bit microprocessors (currently the Motorola 68020 and the AT&T 32100), on board floating point coprocessors, and 2 MBytes of memory. These CPUs are standard VME modules with full master and slave capabilities.

Central to the multiprocessor system is the ACP Branchbus, which is the means for connecting the potentially large numbers of crates that contain the individual processing elements. The farm of processor "nodes" is managed by a single host computer. The nodes run identical FORTRAN programs. They process and analyze separate data samples concurrently. The data is delivered and retrieved by the host and its tape drives and disks.

The design goals of the project emphasized an open and competitive environment for the processors. In this way the system is receptive to the most cost effective processing elements available from industry at any time. The goal was to allow individual processor nodes to function, as appropriate, in any of a number of 32 bit multi master bus systems such as VME, Multi-bus II, NuBus, FASTBUS and VAXBI Bus. With the exception of FASTBUS, none of the buses had a means for connecting several crates together at high data rates. The ACP Branchbus addresses this problem of linking several high performance local buses to a host. FASTBUS was felt to be more complex (and expensive) than required for the immediate and future needs of this application involving single board computers. In addition, the larger VME customer base was more conducive to ultimate commercialization of ACP designs. The ACP Branchbus was therefore designed to allow high speed block transfers to or from a host to any one of a number of slaves residing in any of a number of crates. This bus now forms the backbone of both off line and on line ACP systems.

[1] I.Gaines *et al.*, *The ACP Multiprocessor System at Fermilab*, presented at the Computing in High Energy Physics Conference, Asilomar State Beach, California, February 2-6, 1987. FERMILAB-Conf-87/21; J. Biel *et al.*, *Software for the ACP Multiprocessor System*, presented at the Computing in High Energy Physics Conference, Asilomar State Beach, California, February 2-6, 1987. FERMILAB-Conf-87/22; Detailed module and system descriptions may be found in *Advanced Computer Program: Multiprocessor System Hardware Manual.* High energy physicists with access to DECnet may access an up to date list of all of the ACP's technical manuals, publications, and available software at FNACP::ACPDOCS_ROOT:[DOCS]DOCLIST.DOC .

The ACP Branchbus

Original Implementation

As originally developed, the ACP Branchbus is a 32 bit bus connecting a single master (presently a microVAX or FASTBUS) to multiple crates (presently VME). (Multi master capabilities are presently being tested with a new Branchbus interface card, as will be discussed below.) It is a differential bus (RS485 protocol) and runs at a maximum data rate of 20 MBytes per second. The bus may be up to 50 feet in length, with up to 31 slave crates allowed. Byte parity is implemented on all transfers. Address and data are multiplexed. Physically, the bus is two 50 conductor twist and flat ribbon cables with connectors at each master and slave.

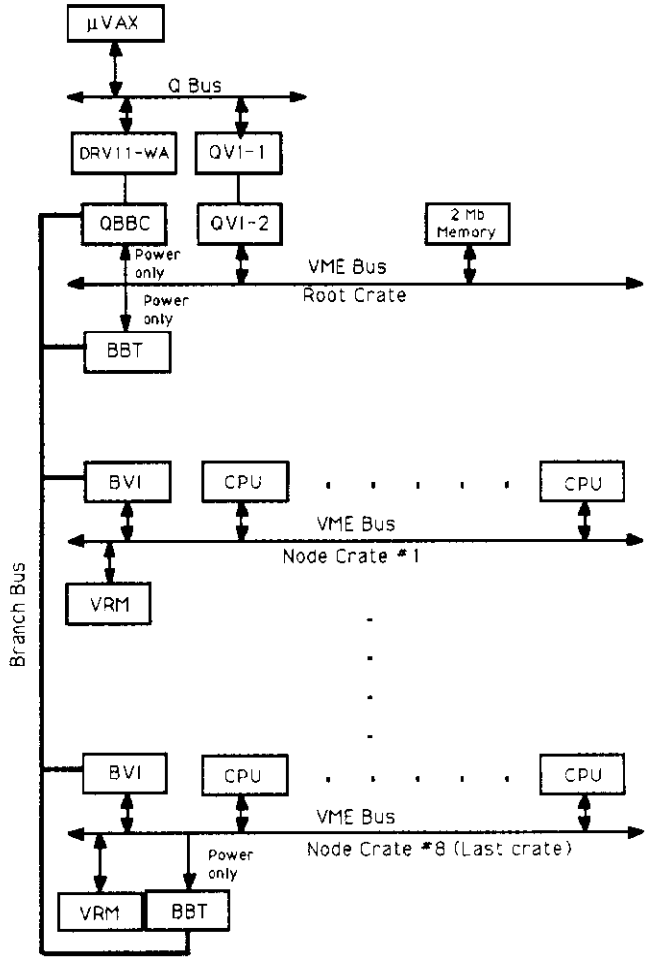

Figure 1. A full scale off line ACP Multiprocessor system as presently installed in the Fermilab Computer Center.

An ACP Branchbus system, typical of existing off line installations, is shown in Fig. 1. Individual modules are discussed in some detail below. A Branchbus Controller (BBC) controls the bus as master. Here, the master is a QBBC which connects to the QBus through a standard DRV11-WA DMA interface. (Unibus masters can also connect to the QBBC through a DR-

11W interface.) Also existing is a FASTBUS interface (FBBC), and in design is a VME interface (VBBC), both discussed below.

Distributed along the Branchbus are Branchbus slaves. In this instance the slaves are interfaces to VME crates of CPUs. Each Branchbus slave acts as a master in its own crate. Up to 31 slaves can be connected to a single branch. Presently supported is the VME bus through a Branchbus to VME Interface (BVI). If warranted, other "x" bus crates could be supported through a BxI module, or individual modules can be interfaced directly to the branch bus as slaves.

Operation of the Branchbus is as follows: Data transfers on the bus consist of a control cycle followed by data cycles. During one half of the control cycle a BBC places on the Branchbus a 32 bit word packed with Branchbus slave address, transfer count and the type of bus transfer the Branchbus slave should perform in its crate. The second half of the control cycle contains the source or destination address in the crate for the data transfer. The ensuing data cycles transfer data between the crate bus and the QBus (or FASTBUS, etc., depending on the type of BBC). Slaves signal acceptance of data or an error condition. Masters may abort block transfers with a reset cycle. Individual word handshaking is not used, but a wait line is provided to prevent data being lost from a "receiver not ready" condition.

Several important module designs that support Branchbus operation are presently in use. These are the QBBC (or UBBC), the BVI, the Branchbus Interface Daughter Board (BBIDB), and the FBBC. These are described in the following. In addition, a system requires a Branchbus Terminator (BBT) for proper signal termination at each end of a Branchbus cable.

QBBC (UBBC): QBus (or Unibus) to Branchbus controller - This module when paired with a standard DRV11-WA module from DEC provides a connection between the microVAX's QBus (or a big VAX's Unibus through a DR11W module) and the Branchbus. It operates as a DMA device on the Q/U bus providing input and output transfer lengths of up to 64 KBytes as either 16 bit or 32 bit words. It supports data transfer rates of 500 KBytes/sec. A switch selectable "burst mode" may be used to double these rates in those installations where the QBus can tolerate burst operation. This module is able to swap the order in which words and bytes are placed in or retrieved from 32 bit Branchbus words. Swapping is under control of 2 bits in the first control word preceding each block. Thus swapping is dynamic and may be changed from block to block.

BVI: Branchbus to VME Interface - This module is the connection between the Branchbus and a VME crate. It is a slave on the Branchbus and a master on the VME bus. Via the BVI, the host computer can write or read any address in a VME crate attached to the Branchbus. The BVI supports data rates, through VME sequential transfers, of better than 20 MBytes/sec.A double high VME module residing in a single slot of a VME crate, it translates protocols between the Branchbus and VME. The Branchbus cables connect directly into the BVI front panel. The following summarizes the features of the BVI:

1. Branchbus slave.
2. VME Master: Single word as A32 or A24, D32 or D16; Sequential as A32 or D32.
3. On output block transfers (Branchbus master to slave), the BVI ignores the transfer count (in the first Branchbus block transfer control word) and writes all the data sent on the Branchbus to VME.
4. On input block transfers (Branchbus slave to master), the BVI reads the number of words specified by the transfer count and sends them on the Branchbus to the master.
5. On board FIFO allows pipelining of output block transfers.
6. Performs single, block, and ACP defined broadcast transfers on VME.
7. Maximum transfer count is 65535 cycles (words or longwords).
8. VME request and grant priority is jumper selectable.

A block diagram of the BVI is shown in Figure 2.

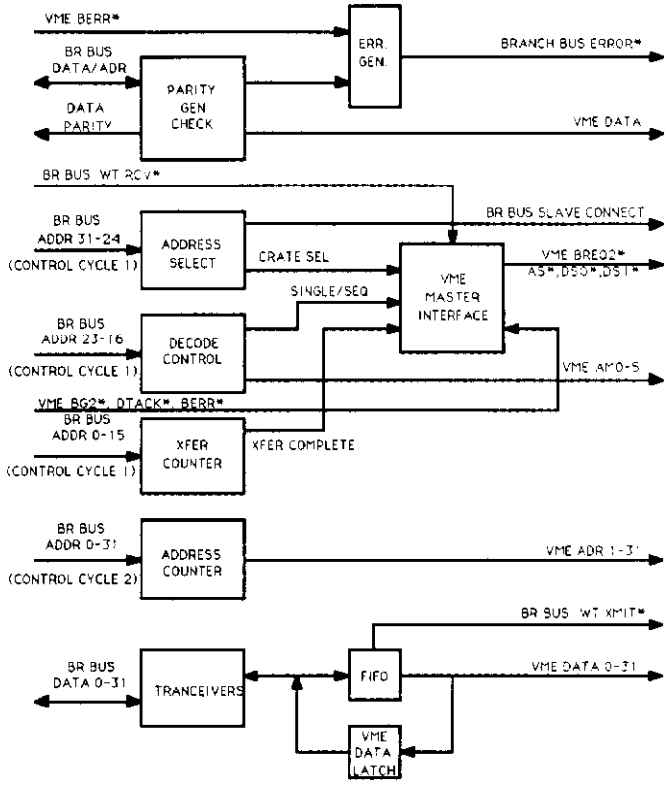

Figure 2. Block diagram of the BVI, a typical Branchbus slave.

BBIDB: Branchbus Interface Daughter Board - This daughter board simplifies the interfacing of various designs to the Branchbus by providing the Branchbus connectors and RS485 transceivers necessary to attach to the high speed, differential Branchbus. The daughter board mounts piggy back to the parent board with the Branchbus connectors extending out through the front panel. The interconnection to the parent board is via single ended TTL through a pin and socket connector. The BBIDB is used in the QBBC and BVI. A new multi master BBIDB is being tested. It may be installed transparently in existing modules and is part of the new VBBC module and Bus Switch Interface Board presently in design and described below.

FBBC: FASTBUS Branchbus Controller - This module, developed by the Collider Detector at Fermilab (CDF) Department in collaboration with the ACP, is a slave on FASTBUS and a Branchbus master. FBBC systems require a FASTBUS master to read or write to the Branchbus through the FBBC. Transfers on Branchbus take place without the FASTBUS master needing to know the specifics of Branchbus protocol. Features of the FBBC include the ability to do pipeline block transfers at better than 200 nsec/word, to do handshake block transfers, and to do random single word reads and writes. Several wire wrapped FBBC modules are in use at Fermilab (CDF Level 3 Trigger[1]) and at Los Alamos (the LAMPF MEGA experiment[2]). Both of these experiments are described later in this paper. The combined FBBC - BVI Branchbus/VME link from FASTBUS into the memory of ACP CPU nodes has been tested, error free for over 48 hours, at 20 MBytes/sec. Lay out of a printed circuit version of the FBBC is nearly finished. It will include multiple master capabilities.

---

[1]B. Flaugher *et al.*, *Integration of the ACP Multiprocessor System into the CDF On Line Data Acquisition Environment*, paper presented at this conference.

[2]M. Oothoudt *et al.*, *Use of the Fermilab Advanced Computer Project (ACP) for MEGA On Line High-level Triggering and Off Line Data Analysis*, MEGA internal document, Los Alamos, November 1, 1985; and M.Oothoudt, private communication.

Two other important designs, which augment those directly involved in Branchbus operation, are in common use in ACP Multiprocessor Systems . These are:

VRM: VME Resource Module - A companion module to the BVI, the VRM provides the standard VME system utilities. It also includes a 32 bit addressable latch that can be written or read over VME. This latch can be used as an attention register for nodes within a crate to signal their status to a host. The VRM also contains a VME bus error register which stores the address of a bus error.

QVI: QBus to VME Interface - In the ACP system, this module provides QBus memory extension by mapping QBus address space to VME address space. The QVI is a single transfer VME master, capable of 8 or 16 bit transfers to short, standard, and extended VME address space. It was initially intended to permit two or more microVAXes to share memory and, thereby, host responsibilities. It clearly has many other applications, not the least of which is to allow a VBBC, described below, to replace a QBBC in future systems. It also allows small scale, single crate, ACP systems to be operated without the Branchbus.

Omnibyte Corporation of West Chicago, IL, presently supplies ACP designed modules including the QBBC, BVI, QVI, VRM, BBT, and the 68020 based CPU module described elsewhere. In addition, crates, power supplies and cooling as well as [correctly] ready made cables may be obtained from Omnibyte.

Work in Progress

The ACP Branchbus has proven to be a simple to use and reliable bus when used in our applications. The ACP is currently working on the following improvements which make the system more versatile:

Multi-master Branch Bus - We have defined a distributed arbitration scheme similar to that used on the SCSI bus to transform the Branchbus into a multi master bus where up to 16 masters can arbitrate for and use the bus. No changes were made to the protocol or definition of the bus other than using one of the previously spare signals and defining the additional arbitration cycle.

Implementing the multi master bus with existing modules is simply a matter of replacing the BBIDB on Branchbus masters with a new Multi Master Branchbus Interface Daughter Board (MMBBIDB) which transparently handles the arbitration. Branchbus slaves will work with the multi master Branchbus without modification. For example, replacing the old BBIDB with a MMBBIDB makes the QBBC is now a master on the multi-master bus. The number of masters on a branch is limited to 16 (with 31 slaves as noted earlier).

This new daughter board design is finished, PC layout is finished, and production units are currently being tested.

VBBC: VME Branchbus Controller - This design will allow any VME processor to become a master on the ACP Branchbus. The VBBC will supplement existing Branchbus controllers. Used with appropriate interfaces to VME, such as the QVI or commercial FASTBUS to VME interfaces, it may also ultimately replace the QBBC and the FBBC. With the multi-mastership capabilities, the VBBC will allow systems like that shown in Fig. 3. In this system, any processor in any crate can take mastership of the Branchbus using the VBBC and communicate with any other processor anywhere in the system. The VBBC also allows convenient use of Branchbus in data acquisition systems, as discussed below.

The VBBC provides a VME to Branchbus interface whereby devices on VME which have VME mastership capability can use the VBBC as a port to write and read to other devices connected to the Branchbus. The device is a VME slave/Branchbus master built on a double high VME board. The Branchbus cables plug into the front panel of the device. It is a shared resource on the VME bus although only one master at a time may use it. It is allocated on a first-come, first-served basis by a test-and-set bit in its control register. Once programmed, the Branchbus cycles occur transparently to the VME master which is writing data to or reading data from the VBBC. A block diagram of the VBBC is shown in Figure 4.

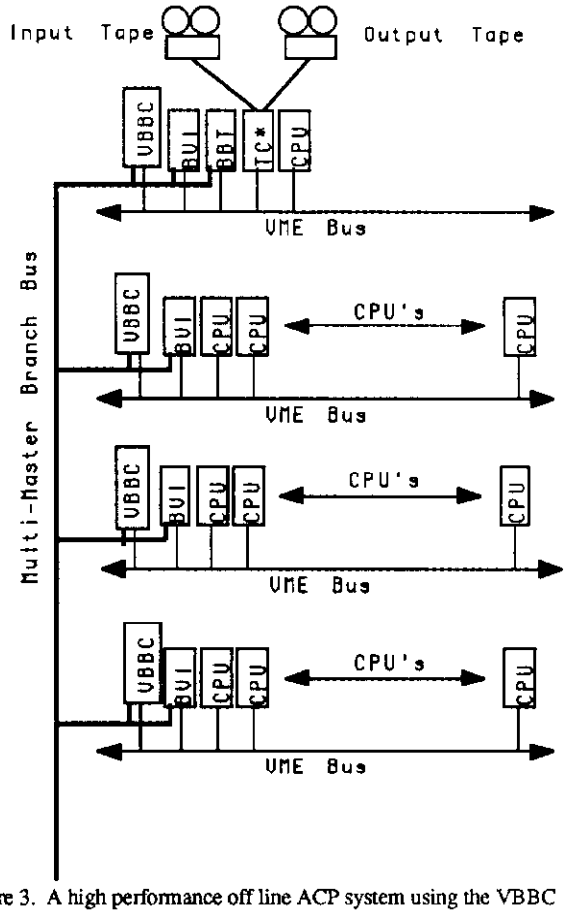

Figure 3. A high performance off line ACP system using the VBBC

An attention register and optional VME bus system controller are included on the module which allow VBBCs to perform the additional functions of the ACP VME Resource Module (VRM) in ACP systems thus saving a slot in the VME crate.

In an ACP system, a VME master such as a QVI or a CPU can orchestrate data transfers directly between the VBBC and a tape controller. The master would set up and trigger a VBBC transfer to or from some Branchbus address. Then it would set up and trigger a tape controller transfer to or from the data port on the VBBC. The data transfer would then take place with the tape controller acting as a DMA device. Unlike with the QBBC, the tape I/O data rates are not limited by QBus nor is the Branchbus transfer limited to the 500 Kbytes/sec of the DRV-11WA. In addition, the VBBC can replace the VRM in the Branchbus crates as a VME system controller and as a central crate "mailbox" module.

Features of the VBBC are:

1. VMEbus Slave: Single or sequential transfers, A32, A24, D32, D16.
2. Bus master in single-master or multi-master Branchbus systems.
3. VMEbus System Controller option (Switch selectable)
4. 24 Bit Attention Register (identical to the VRM)
5. On board FIFOs to buffer data received from Branchbus during Input Transfers
6. 32K Maximum Transfer Count (16 or 32 Bit Words)
7. Maximum Transfer Rate of at least 20 Mbytes/sec
8. Programmable Byte and Word Swapping Capability (Identical to the QBBC).

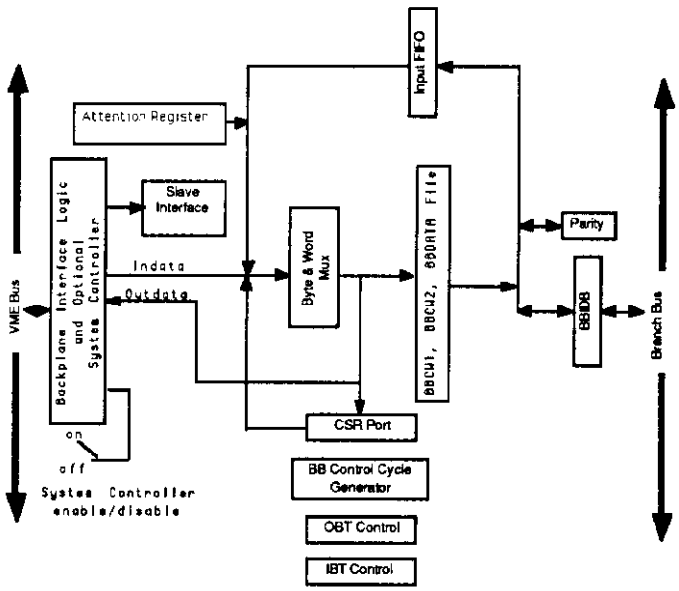

Figure 4. Block diagram of the VBBC, an important Branchbus master, presently in development.

The ACP Branchbus Switch - Design is also under way on a 16 branch crossbar that allows up to 16 multi master Branchbuses to be interconnected. With this switch, any device which can become a master on Branchbus can connect to any slave in the entire switch connected system. All channels of the switch can be active at any time. For example eight of the Branchbuses could be connected to the other eight, all transferring data simultaneously for an aggregate inter crate data rate of 8 X 20 MBytes/sec or 160 MBytes/sec. This is in addition to the local bus activity present in the VME crates which are not part of the system transfers.

The Switch is based on the Texas Instruments 74AS8840 16x16 four bit crossbar switch chip which has a maximum propagation time of 25 nsec. The Switch is a backplane incorporating 14 of these chips. Modules may be plugged into the Switch crate (Figure 5) much as with VME or Multibus. However, instead of signals being interconnected in a bus structure, each connector is a crossbar switch point.

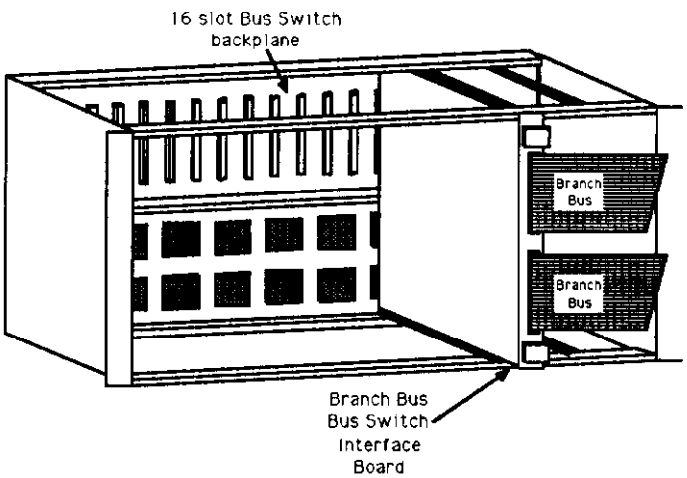

Figure 5. The ACP Branchbus Switch. The backplane uses single ended TTL Branchbus protocol.

A TTL compatible version of Multi Master ACP Branch Bus is used as the communications protocol. This protocol lends itself well to block transfer operations. Each port has a bandwidth of at least 20 MBytes/sec using LS/ALS technology. A Branchbus Switch Interface Board (BSIB) module is designed which plugs into the crate and converts the TTL Branchbus to standard differential ACP Branchbus protocols. In this way Branches with masters and slaves on them can communicate with other branches in a transparent fashion. The crate hardware is Eurocard standard, 220 mm deep, accommodating 6U and 3U high modules. The BSIB module is expected to be a 6U high board. It is also possible, as described below, to plug active modules, such as special CPUs, directly into the Switch Crate when appropriate.

The Switch makes possible both higher performance on line systems described below as well as off line systems with very large numbers of nodes. An example of such a system, based on one being developed for theoretical physics use, is shown in Fig. 6. In this example, 128 Switch Crates each containing eight directly connected floating point processor modules are combined to create a system with a total of 1024 nodes. All of these can be communicating with other nodes in the system simultaneously at 20 MBytes/sec between each pair (aggregate over 10 GBytes/sec!). The interconnections between Switch Crates are through the BSIB Modules and the standard ACP Branch Bus.

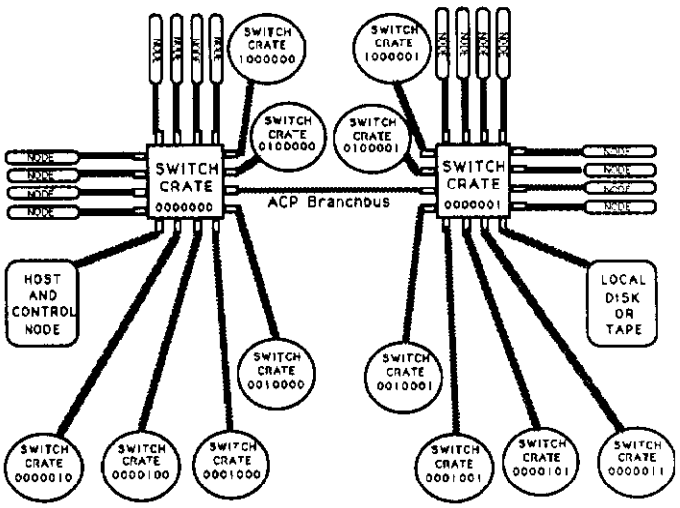

Figure 6. A 1024 node floating point processor built around the Branchbus Switch

On Line Applications of the ACP Multiprocessor System

The flexibility, ease of use, and low cost of the ACP system have made it attractive for use in on line applications requiring large amounts of processing power. The speed, reliability and inter rack distance transmission capability of the ACP Branchbus make it the natural way to interface a system of ACP processors to a data acquisition system. The different Branchbus modules described in the preceding section allow Branchbus systems to be interfaced directly to Unibus, QBus, VME bus, and FASTBUS. We will review here significant on line applications of the ACP system by various groups collaborating with the ACP. The main focus is on the use of ACP systems in conjunction with FASTBUS based data acquisition systems.

The Collider Detector at Fermilab (CDF) uses an ACP multiprocessor system, now planned to be over 100 VAX equivalents in processing power, as a high level trigger processor. Figure 7 shows how the ACP system, including a microVAX host, is fully integrated into CDF's FASTBUS data acquisition system. The FBBC, a FASTBUS slave and Branchbus master, acts as the interface between the FASTBUS system and the Branchbus. FASTBUS event builders use FBBCs to send events to the processors. They are under control of the CDF Buffer Manager which manages the flow of

events through the data acquisition system and keeps track of the availability of nodes. Processed events are read out by an additional FBBC controlled by a system of several large VAXes. In these "consumer VAXes" the events can be made available to any of a number of on line processes. These include the data logger that writes output tape as well as monitoring and analysis tasks.

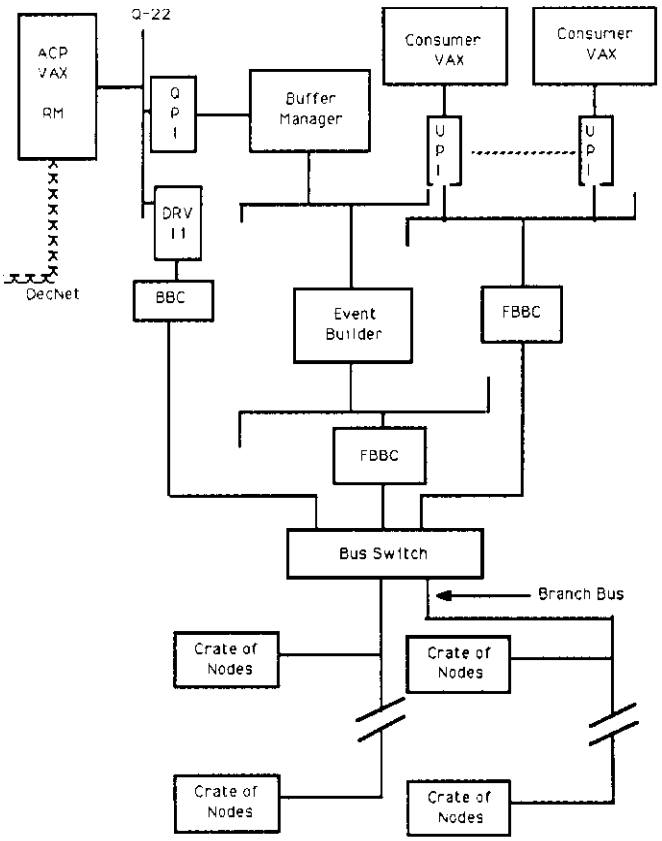

Figure 7. Block diagram of the CDF data acquisition system

A microVAX host runs the same ACP software as is used in off line applications. It downloads programs and calibration constants to the processing nodes, monitors node performance, processes exceptions, and sums up statistics. This microVAX polls the processing nodes to detect event completion and reads out 128 bits of software trigger bits from each node that has completed an event. This data is delivered to the Buffer Manager over FASTBUS allowing the Buffer Manager to determine, based on requests made by any consumer process, whether the event should be read out or whether the processor should be immediately returned to the list of available processors.

Thus, in this application, both the Buffer Manager and the ACP host are involved in every event: the Buffer Manager tells an event builder where in the multiprocessor to deliver the event; the ACP host detects event completion; and the Buffer Manager determines the disposition of the event based on data supplied by the ACP host. This requires the Buffer Manager and ACP host to communicate (over FASTBUS) several times per event, which limits the event throughput of such a system to roughly 100 events per second. This limit is acceptable here since other aspects of the data acquisition impose limits at a similar level.

The MEGA experiment at the Los Alamos Meson Physics Facility uses an alternative method of integrating an ACP Multiprocessor System into a FASTBUS environment. Here, as in CDF, events are delivered to the processors through an FBBC. However, in MEGA the processors themselves know the criteria for a successful event, and the ACP host needs only deal with the rare event that passes the trigger algorithm resident in the processing nodes. The vast majority of events which fail the trigger requirement are managed directly from the FASTBUS which detects that the node has completed processing through the FBBC. The events are combined into buffers so that many events at once are delivered to an individual processing node. These techniques allow the experiment to process tens of thousands of events per second. This is appropriate here since, in contrast to CDF, the events are simpler and require much less processing.

Another experiment interfacing Branchbus to a FASTBUS system is Fermilab E687, an experiment in the wide band photon beam. Here an ACP system is being added to an existing FASTBUS/PDP-11 data acquisition system. Initially, a small ACP system will be used to provide additional computing power for experiment monitoring using a subset of events. Thus, events will be delivered to the Branchbus system through an FBBC by a GPM FASTBUS master acting as a parasite in the data acquisition crate. Since the GPM/Branchbus system can read out events much faster than the existing Unibus interface, the fraction of events that the ACP system can handle is determined simply by how many nodes are installed in the system. At a later stage, when more nodes are available, the Branchbus will become the primary data acquisition path, with all events being shipped to the ACP processors, allowing the system to be used either as a high level trigger filter or as the first stage of off line analysis.

A non FASTBUS on line application of ACP processors is Fermilab E769, a hadron experiment in the Tagged Photon Laboratory. Here, Branchbus is used merely as a VAX to VME interface, allowing ACP processors to be downloaded from the VAX. The data acquisition system is VME based. Specially designed CAMAC controllers deliver CAMAC data in parallel into multiple VME memory modules, from where full events are assembled and processed by the ACP CPUs. Output tape is also written directly from VME. In a similar approach, M.Levine has developed a system for a Brookhaven heavy ion experiment which doesn't use Branchbus at all. Instead, a specially designed VME to FASTBUS interface delivers data to the ACP processors, and a commercial VAX to VME link is used to read out the events.

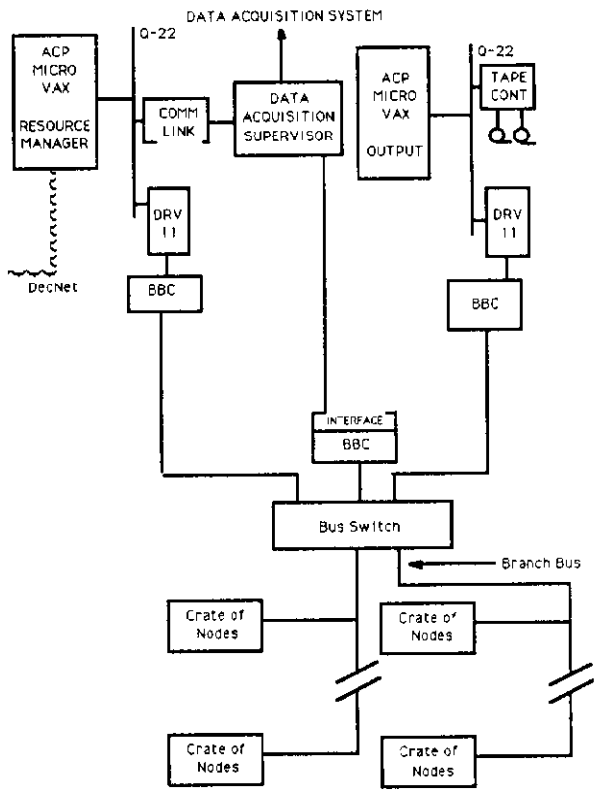

Figure 8. Diagram of a data acquisition system interfaced to Branchbus.

In addition to the actual experiment applications we have just described, Figures 8 and 9 suggest other ways of interfacing branch bus systems to data acquisition systems. Fig. 8 shows a system similar to CDF, where a data acquisition supervisor delivers events into the branch bus. The only new module needed is a DA bus to Branchbus interface. If the DA bus is either VME or FASTBUS one may take advantage of the VBBC or FBBC for interfacing.

A very attractive opportunity for obtaining a well integrated data acquisition system and high level trigger exists when the digitization electronics are primarily in VME. In this situation, as shown in Figure 9, the ACP Branchbus is ideally suited to be the inter crate data acquisition read out bus in addition to its usual role interconnecting crates of ACP CPUs. The hardware and software at both ends of this system are identical resulting in major economies in expertise and support required. The Data Acquisition crates may contain ACP CPUs that can be used for data formatting, monitoring, and calibration. These communicate with the host using standard ACP support software subroutines. It is also possible to write directly to tape using VME based tape controllers, such as that made by Ciprico which is capable of reading and writing tape data directly to or from a CPU in any VME crate through the Branchbus-VME link.

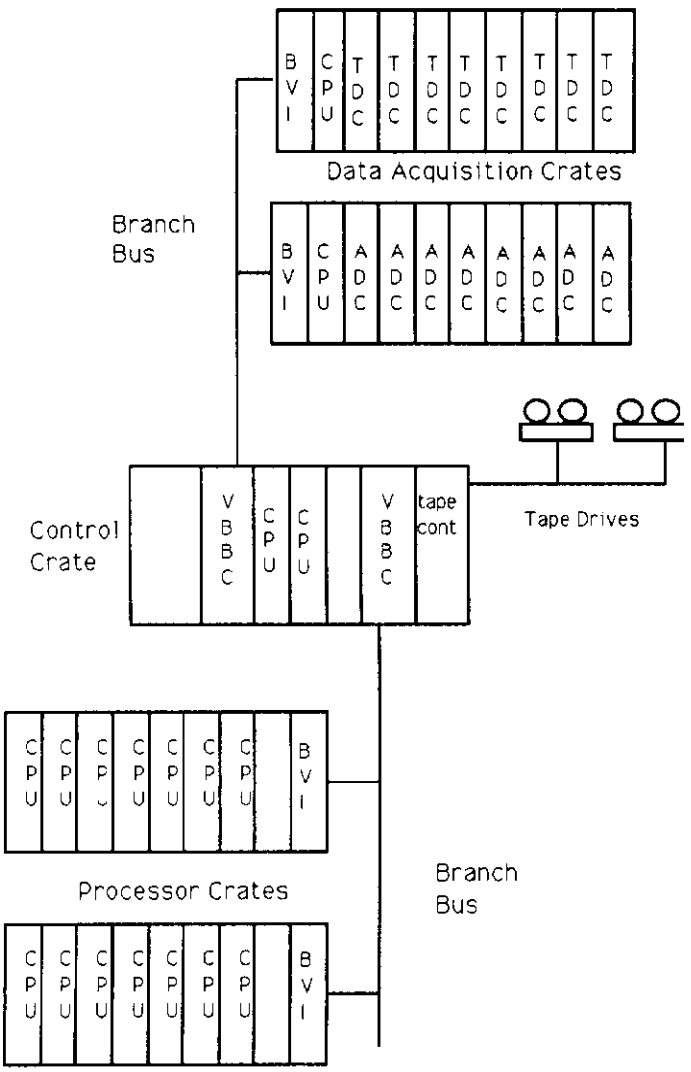

Figure 9. Block diagram of a VBBC based VME data acquisition and processing system.

The Future

Two exciting new product technologies from industry are making it possible for the ACP to start development on still more productive enhancements of the Multiprocessor System. One of these technologies is so called reduced instruction set computers (RISC) now becoming available as microprocessor chip sets. We have started design on a new ACP CPU based on such a product. Reconstruction code benchmarks indicate that these new CPUs will run 10 times faster than the present ACP CPUs.

The other development is the extremely low cost and high density application of video recording technology for recording data. Most promising appears to be cassette recorders that can store 2 GBytes of data (10 times a 2400 ft 6250 BPI tape) on a $10 cassette, reading and writing at speeds approaching that of standard tape drives. Since these devices are priced at $2000 each in quantity and come with SCSI interfaces, they have two natural applications in the ACP System environment. Attaching one each to ACP CPUs they can be used in large numbers for rapid parallel scanning and analysis of huge reconstructed event data basis. On line groups of these CPU-cassette recorder combinations can write raw data in parallel. One can readily anticipate an order of magnitude increase in data writing rates in this way. We are expecting a first unit to arrive in June for testing.

Acknowledgements

We acknowledge contributions that S. Bracker and G. Case made to various aspects of the ACP Branchbus system. M. Larwill designed the FBBC. T. Carroll, T. Devlin, B. Flaugher, and U. Joshi participated in system tests of the FASTBUS to ACP CPU link.